

### AMENDMENTS TO THE SPECIFICATION

#### In the Specification:

Please replace the paragraph starting at line 5 on page 7 with the following replacement paragraph:

Turning briefly to Fig. 5, a driver component 410 is illustrated in further detail in accordance with an aspect of the subject invention. Driver component 410 comprises driver code 412 and import address table 414. Driver code 412 corresponds to software specified procedures and functions that driver component 410 utilizes to implement driver functionality. Driver component 410 can enhance its utility while minimizing its overall size by using external services or procedures 422 (Fig. 4) provided by the kernel component. Driver component 410 accesses external procedures 422 by “importing” them using an import address table 414. When a driver component 410 is loaded or executed the procedures 422 listed in the import table can be bound to the drivers so that the driver can utilize the functions and procedures provided therein. This binding is referred to herein as driver linkage or import linkage.

Please replace the paragraph starting at line 16 on page 7 with the following replacement paragraph:

It should be noted that while this detailed description focuses almost exclusively on drivers and driver components, the scope of the invention is not so limited. The scope of the present invention covers any applications or components, drivers or otherwise that are capable of being shimmed. While this description focuses on drivers and driver components, it is not meant to exclude of all other software applications capable of being shimmed, but rather to facilitate a clear and understandable description of the invention devoid of confusing terms (*e.g.*, client/application/component/driver).

Please replace the paragraph starting at line 8 on page 10 with the following replacement paragraph:

Driver Y 720 goes through a similar process with the common shim component 750. In this case, however, driver Y 720 utilizes its own context component 740 and a different procedure call. Driver Y 720 ultimately seeks to call Foo procedure 762, but as shown here after calling VerifierFoo procedure 764. The hook component 742 of the driver context component 740 determines and saves information related to the procedure called by the shim component 750. Thunk component 744 ~~the~~ links the driver to the shim component 750 by changing ~~is~~ the reference address in driver Y's import address table 724. Thereafter, when driver Y 720 calls the Foo procedure in its code section 722 control is transferred to the context component 740. Context component then stores the stores a pointer to the context information in a register or by alternative means transfers the location of the context information or the context information itself to shim component 750. Shim component 750 then executes its functionality and then using the context information jumps to the verifier procedure 764, which executes and jumps to the Foo procedure 762 that is associated with an oskrnl.exe code at 760. It should be noted that by retrieving and maintaining context data for each driver, the subject invention ensures that context data for previously shimmed drivers is not lost upon the utilization or calling of the shim by another driver. Conventional shimming practices would have lost information regarding the context of the driver X 710 upon execution of driver Y 720. In this case, if driver X 710 was called again after driver Y 720, a conventional shim would not know which procedure driver X 710 originally referenced as context data would not have been retained and the reference in the drivers import address table would have been changed to reference the shim component 750. The present invention eliminates this problem by storing unique context information for each driver component.

Please replace the paragraph starting at line 22 on page 10 with the following replacement paragraph:

System 800 can also include a diagnostic component 840 that analyzes a computer system and initiates corrective action. According to one aspect of the subject invention the diagnostic component 840 assists in root cause analysis of a system crash. The diagnostic component can therefore employ a variety of methods of analyzing system dump information and/or a program trace (*e.g.*, utilizing expert systems, neural networks...) to determine the cause of the crash. However, diagnostic component 840 need not wait for a system crash. The diagnostic component 840 can also be proactive and engage in system analysis to detect system instabilities and/or inefficiencies. Upon determining the cause of a crash or detecting system instabilities and/or inefficiencies, corrective action can be initiated by the diagnostic component 840. Such corrective action can include providing notification to a user or operator *via* interface 850. Corrective action can also comprise searching the shim database 830 to determine whether a shim component already exists that if applied can fix the detected problem, instability or inefficiency. Such a determination can be made intelligently using Bayesian networks, neural networks, decision trees, support vector machines, linear and non-linear regression and/or other learning models. According to an aspect of the invention, the diagnostic component can engage in a probabilistic analysis based on the cost of making an incorrect diagnosis and/or selecting the wrong shim weighed against the benefit of correction. Confidence levels may be employed and specified by a developer or user to control actions of the diagnostic component.

Please replace the paragraph starting at line 11 on page 16 with the following replacement paragraph:

Computer 1212 also includes removable/non-removable, volatile/non-volatile computer storage media. Fig. 12 illustrates, for example disk storage 1224. Disk storage ~~4124~~ 1224 includes, but is not limited to, devices like a magnetic disk drive, floppy disk drive, tape drive, Jaz drive, Zip drive, LS-100 drive, flash memory card, or memory stick. In addition, disk storage 1224 can include storage media separately or in combination with other storage media including, but not limited to, an optical disk drive such as a compact disk ROM device (CD-ROM), CD recordable drive (CD-R Drive), CD rewritable drive

(CD-RW Drive) or a digital versatile disk ROM drive (DVD-ROM). To facilitate connection of the disk storage devices 1224 to the system bus 1218, a removable or non-removable interface is typically used such as interface 1226.